

Der Einsatz von Big Data und Verkehrssimulation SUMO im Rahmen von Rail2X

The use of big data and the SUMO transportation simulation in Rail2X

Sangeetha Shankar | Lucas Andreas Schubert | Akhil Jayant Patil | Jakob Erdmann

Das Ziel des Projektes Rail2X – „Smart Services“ ist die Erprobung der Kommunikationstechnik Vehicle-2-Everything (V2X) im Bahnbereich, in dem drei Anwendungsfälle im Feld aufgebaut und demonstriert werden. Ein Big-Data-Cluster bestehend aus diversen Open-Source-Big-Data-Technologien wurde aufgebaut, um Daten aus den Anwendungsfällen zu speichern, zu prozessieren und zu analysieren. Zusätzlich wurde die Simulationssoftware SUMO verwendet, um die Anwendungsfälle großskaliert zu demonstrieren. Dieser Beitrag beschreibt den technischen Aufbau des Big-Data-Clusters sowie die Simulationsumgebung, um die drei Anwendungsfälle in einer extrapolierten Region zu simulieren.

1 Einführung

Ziel des Projekts Rail2X – „Smart Services“ [1] ist die Erprobung des drahtlosen Standards IEEE 802.11p im Bahnbereich. Dieser Standard wird im Automobilbereich in der Kommunikation Vehicle-2-Vehicle (V2V), Vehicle-2-Infrastructure (V2I), oder noch allgemeiner, Vehicle-2-Everything (V2X) erprobt. Der Standard verwendet den 5.9-GHz-Frequenzbereich und hat eine Reichweite von ein paar hundert Metern. Um diesen Standard im Bahnkontext zu erproben, wurden drei unterschiedliche Anwendungsfälle auf der 24 km langen Strecke zwischen Annaberg-Buchholz und Schwarzenberg im Erzgebirge aufgebaut und demonstriert.

Der erste Anwendungsfall, genannt „Service und Diagnose“, besteht aus der Idee, dass ein Zug Daten für Service- und Diagnosezwecke aus Infrastrukturelementen auf der Strecke über V2X sammelt und sie an ein sogenanntes Depot übermittelt, welches wiederum die Daten in das aufgebaute Datenmanagementsystem überträgt. Zu diesen Daten gehören unter anderem Wetterdaten einer aufgebauten Wetterstation und Temperaturdaten einer Weiche. Dieser Anwendungsfall ist besonders interessant in Regionen wo keine Mobilfunkabdeckung vorhanden ist. Der zweite Anwendungsfall, genannt „Anrufschränke“, beschäftigt sich mit der Erhöhung des Fahrerkomforts durch die Anwendung von V2X in der Kommunikation zwischen einem Straßenfahrzeug und einem Bahnübergang (BÜ) beim Überquerungsprozess. Der Komfort wird dadurch erhöht, dass ein Autofahrer den BÜ-Querungswunsch per Knopfdruck an einem Taster signalisieren kann. Dadurch wird die entsprechende V2X-Nachricht an den BÜ gesendet. Im Straßenfahrzeug wird der BÜ-Status, „Schränke auf“ oder „Schränke zu“ kontinuierlich visualisiert. Der dritte Anwendungsfall, genannt „Bedarfshalt“, beschäftigt sich mit dem Austausch von Daten zwischen einem Zug und einem Bahnhof. Bei diesem Anwendungsfall kann entweder ein Passagier im Zug oder ein Gast im Bahnhof einen Bedarfshaltstaster betätigen, sodass der Zug im Bahnhof auch anhält, wenn es die betriebliche Situation ermöglicht. Um

The Rail2X – „Smart Services“ project aims to evaluate the use of Vehicle-2-Everything (V2X) communication in the railway sector via three use cases. A Rail2X demonstrator has been established in the field for this purpose. A big data cluster consisting of several open-source big data technologies has been established to store, process and analyse the data pertaining to the use cases, which is collected from different entities in the railway network. The SUMO simulation software is used to demonstrate the concepts on a large scale. This article explains in detail the setup of the big data cluster as well as the simulation environment to simulate the three use cases in a larger region.

1 Introduction

The Rail2X „Smart Services“ project [1] focuses on evaluating the use of the IEEE 802.11p wireless standard in the railway sector. This standard is being tested and evaluated in the automotive sector in use cases such as Vehicle-2-Vehicle (V2V), Vehicle-2-Infrastructure (V2I) or, in a more general case, Vehicle-2-Everything (V2X) communication. The standard uses the 5.9 GHz band and has a range of several hundred metres. Three different use cases have been defined and established on a 24 km stretch of track in the Erzgebirgsbahn, between Annaberg-Buchholz and Schwarzenberg/Erzgebirge, in order to evaluate the use of this standard in the railway sector.

The first use case, defined as „Service and Diagnosis“, consists of using a train to collect service and diagnosis data from infrastructure elements such as switches and weather stations using V2X and to then transport the collected data to a depot, where it is uploaded to the data management system. This use case is especially interesting in regions with low or no mobile signal coverage. The second use case, defined as „Level Crossing“, focuses on increasing the comfort in the use of railway crossings via the V2X standard in order to enable communication between a car and a level crossing (LC) during the process of driving through a LC. The comfort for the car driver is increased, as the driver can press a button inside the car to send a LC cross request via a V2X message. The „level crossing closed“ or „level crossing open“ status can be continually visualised inside the car. The last use case, defined as „Request Train Stop“, focuses on the exchange of information between a train and a train station by means of the sending of a „train stop“ request from the train to the station or vice-versa. In this use case, a hopping station has been used to increase the communication range between the approaching train and the train station. The hopping station was built a few

Rail2X Projektkonsortium Rail2X Project Partners	u.a. mit folgenden Projektaufgaben / Expertisen with the following project tasks / expertises	 
DB Systel GmbH DB Systel GmbH	als Konsortialführer mit Expertise in Eisenbahn-IT as consortium leader and expert in railway IT	
DB RegioNetz Infrastruktur GmbH Erzgebirgsbahn DB RegioNetz Infrastruktur GmbH Erzgebirgsbahn	Bereitstellung des Testfeld der Erzgebirgsbahn für die Demonstration provides the test field for the demonstration	
Siemens Mobility GmbH Siemens Mobility GmbH	Expertise in der Steuerungs- und Sicherungstechnik experience in control and security technology	
Hasso Plattner-Institut der Universität Potsdam Hasso Plattner Institute of the University of Potsdam	Wissenschaftliche Expertise in IT-Sicherheit scientific expertise in IT security	
DRALLE Systementwicklungen DRALLE system developments	Erfahrung bei der Implementierung von Demonstratoren experience in the implementation of demonstrators	
Deutsche Zentrum für Luft- und Raumfahrt e.V. German Aerospace Center (DLR)	wissenschaftlichen Expertise im Eisenbahnwesen scientific expertise in railway engineering	

Tab. 1: Kompetenzen des Rail2X-Projekt-Konsortiums

Tab. 1: The competencies of the Rail2X project

Quelle / Source: Rail2X

die Kommunikationsreichweite zwischen Zug und Bahnhof zu erhöhen, wurde eine Hopping-Station im Feld aufgebaut. Diese Hopping-Station wurde neben der Strecke ein paar hundert Meter vor dem Bahnhof aufgebaut und leitet die V2X-Nachrichten des Zugs Richtung Bahnhof, und vice versa, weiter.

Die drei kurz beschriebenen Anwendungsfälle wurden im Feld aufgebaut, erprobt und demonstriert. Die Frage ist: Wie können die Anwendungsfälle und deren Nutzung in eine andere geographische Region oder sogar eine größere Region übertragen werden? Eine mögliche Antwort kann durch die Nutzung einer Simulationsumgebung gefunden werden. Im Rail2X-Projekt wurde eine Simulationssoftware in das Datenmanagement integriert, um die drei Anwendungsfälle für ganz Deutschland zu extrapolieren. Der Fokus liegt auf der Nutzung von Open-Source-Lösungen wie SUMO [Abschnitt 3] und Open Data wie OpenStreetMap (OSM) [2], um die Anwendungsfälle großskaliert zu simulieren und zu analysieren. Ein großer Vorteil des Simulierens der Anwendungsfälle ist die Möglichkeit, bestimmte Parameter, wie die simulierte V2X-Reichweite sowie die Position der Hopping-Stationen, zu ändern, um die Ergebnisse aus der Simulation in einer Testumgebung zu evaluieren.

2 Big-Data-Cluster

Das Big-Data-Cluster, das für das Rail2X-Projekt aufgebaut wurde, besteht aus vielen Open-Source-Komponenten (Bild 1). Die Daten von Infrastrukturelementen und Fahrzeugen, sowohl aus dem Feldtest als auch aus der Simulation, werden im Big-Data-Cluster gespeichert und verarbeitet. Es ist vorgesehen, das Big-Data-Cluster nicht nur für dieses Projekt, sondern auch für andere Forschungsaktivitäten im Verkehrsbereich zu benutzen, bei denen große und komplexe Datenmengen verarbeitet werden. Die erste Komponente des Clusters ist Apache Kafka [3]. Das ist ein Framework für die Speicherung und Verarbeitung von Datenströmen, wie z.B. von generierten Nachrichten aus den Anwendungsfällen. Diese Nachrichten sind unter verschiedenen Topics kategorisiert. Anwendungen können im Kafka-Cluster Nachrichten schreiben (Kafka Producer) oder Nachrichten aus dem Kafka-Cluster lesen und verarbeiten (Kafka Consumer). Apache Flink [4] ist ein Stream-Prozessierungsframework. Flink-Anwendungen können Nachrichten von einem oder mehreren kontinuierlichen Datenströmen verarbeiten. Apache Spark [5] ist ein Cluster-Computing-Framework und besteht aus Spark SQL, MLlib, GraphX und Spark-Streaming. Apache Hadoop [6] ist eine Big-Data-Technologie für verteilte Verarbeitung und besteht aus Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN, Hadoop MapReduce und Hadoop Ozone. Apache HBase [7] ist eine skalierbare, verteilte Datenbank und ermöglicht die Verwaltung großer Tabellen. GeoMesa [8] kann umfangreiche, raumzeitliche Analysen mit Daten von Apache

hundred metres ahead of the train station, so that it could relay the V2X messages from the train to the station and vice-versa. These three use cases have been built in the field for testing. The question is: how is it possible to extrapolate the application of these use cases to a different or bigger area? One answer to this question involves the use of a simulation framework to test it. A simulation framework has been integrated into the Rail2X project's data management system in order to simulate these use cases for the entire area of Germany. We have focused on using open-source software, such as SUMO [Section 3], and open-source data, such as OpenStreetMap (OSM) [2], to simulate and analyse the data on a larger scale. A major advantage of simulating the use cases is the possibility of changing the parameters such as the V2X range or the position of the hopping stations and evaluating the simulated results in a test environment.

2 The big data cluster

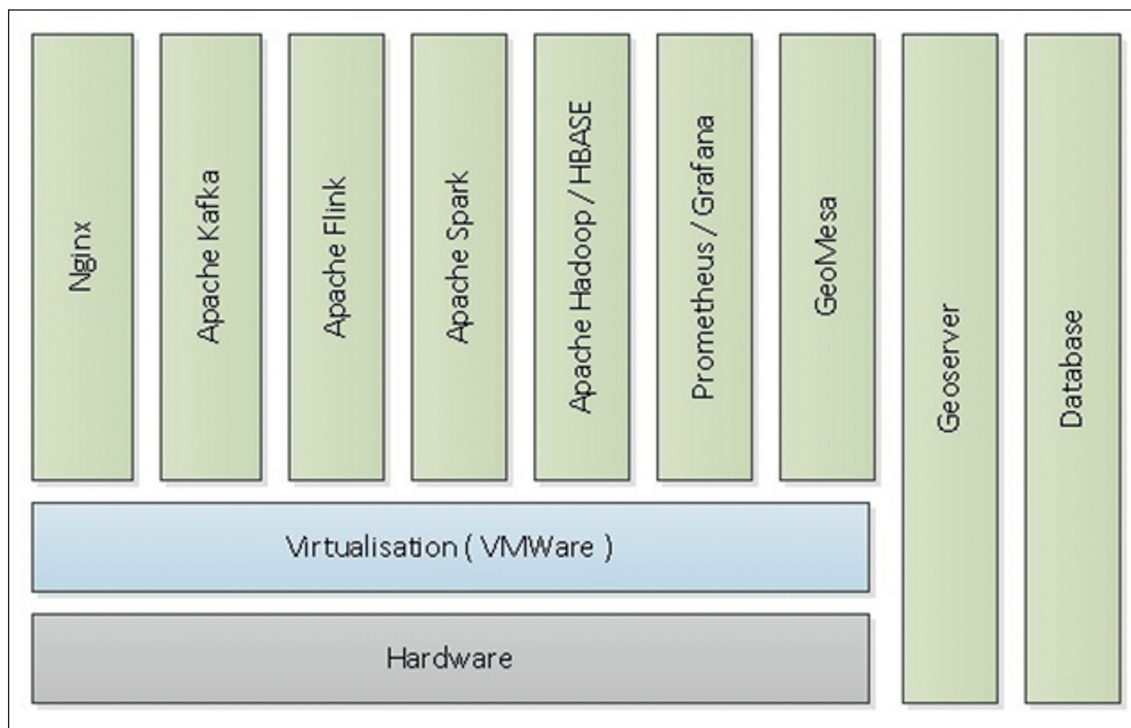
The big data cluster established for the Rail2X project consists of a set of open-source components as shown in fig. 1. The data collected from the infrastructure elements and vehicles in the field as well as in the simulation, is stored and processed in the big data cluster. The cluster is not only intended to be used for this project, but also for other future research activities in the transportation domain involving the handling of large, complex datasets and algorithms, where the use of the big data cluster would be beneficial.

The first component in the cluster, Apache Kafka [3], is a platform to store streams of messages, such as the messages generated from the three use cases. These messages are categorised into different topics. Applications can write messages into the Kafka cluster (Kafka Producers) or read and process the messages from the Kafka cluster (Kafka Consumers). Apache Flink [4] is a stream processing framework. Flink applications can process messages from one or more continuous data streams. Apache Spark [5] is a cluster computing framework, which also includes tools such as Spark SQL, MLlib, GraphX and Spark Streaming. Apache Hadoop [6] is big data technology for distributed processing, which comprises modules such as Hadoop Common, Hadoop Distributed File System (HDFS), Hadoop YARN, Hadoop MapReduce and Hadoop Ozone. Apache HBase [7] is a scalable distributed database designed to host huge tables. GeoMesa [8] is a tool suite which is capable of performing large-scale spatial-temporal analyses of the data stored in Apache Accumulo, Apache HBase, Apache Cassandra and Google Bigtable. In our cluster, GeoMesa is connected to Apache HBase, which in turn is configured to run on top of Hadoop HDFS. A GeoServer [9] is connected to GeoMesa and publishes the data stored in the

Homepageveröffentlichung unbefristet genehmigt für Besucher der Seiten
Rechte für einzelne Downloads und Ausdrücke für Besucher der Seiten
genehmigt von DW Media Group GmbH 2020

Bild 1: Komponenten des Big-Data-Clusters

Fig. 1: The big data cluster – the components Quelle / Source: DLR



Accumulo, Apache HBase, Apache Cassandra und Google Bigtable ausführen. In unserem Cluster ist GeoMesa mit Apache HBase gekoppelt, welches wiederum auf HDFS läuft. Ein GeoServer [9] ist mit GeoMesa verbunden und stellt sowohl die Daten aus dem Cluster als auch die Ergebnisse von GeoMesa-Analysen durch Standard-Dienste des Open GeoSpatial Consortium (OGC) [10], wie z.B. Web Mapping Service (WMS) und Web Feature Service (WFS), zur Verfügung. Prometheus [11] ist eine Zeitreihendatenbank, die gekoppelt mit der graphischen Oberfläche Grafana [12] das Monitoring der oben genannten Frameworks zur Systemoptimierung ermöglicht.

3 SUMO

SUMO ist ein Softwarepaket für die mikroskopische Verkehrssimulation. Das Simulationsmodell bildet die Bewegung von Einzelfahrzeugen und Personen mit diskreten Zeitschritten und kontinuierlichen Ortskoordinaten ab. Die Simulation umfasst sowohl den motorisierten Individualverkehr, Radverkehr und Fußverkehr als auch den Frachtverkehr sowie den öffentlichen Straßen- und Schienenverkehr. Vom Umfang der räumlichen Betrachtung werden einzelne Verkehrsknoten bis hin zu ganzen Städten unterstützt. Eine reine Eisenbahnsimulation von großen Gebieten, wie beispielsweise Deutschland, ist ebenfalls möglich [13].

Die kostenfreie und quelloffene Software bietet große Vorteile bei der Erforschung neuer Technologien und Verkehrskonzepte. Forscher und Entwickler können nach Bedarf Änderungen und Erweiterungen vornehmen, wenn die Möglichkeiten von kommerziellen Werkzeugen erschöpft sind. In der Bahndomäne können mit SUMO neue Betriebskonzepte, Fahrstrategien und Assistenzsysteme entwickelt und getestet werden. Dank der modularen Architektur können leicht neue Modelle integriert werden und kann somit eine Vergleichbarkeit verschiedener Modelle hergestellt werden. Darüber hinaus erleichtert die freie Verfügbarkeit des Quellcodes auch eine institutionsübergreifende Zusammenarbeit mit dem Ziel der Entwicklung einer gemeinsamen Simulationsplattform. SUMO ist dafür unter der Eclipse Public License V2 sowie un-

cluster, as well as the results of the GeoMesa analysis tools using Open Geospatial Consortium (OGC) standards [10], such as the Web Mapping Service (WMS) and Web Feature Service (WFS). The Prometheus monitoring framework [11] is used to monitor the status and performance of the big data cluster. The query and visualisation capabilities are provided by Grafana [12].

3 SUMO

SUMO is a microscopic, space-continuous and time-discrete traffic flow simulation platform. It models the movement of people, vehicles and freight across different modes of transport. This includes the simulation of passenger cars, trucks, pedestrians and bikes as well as rail and road-bound public transport. It can simulate anything from single junctions up to large-scale, city-wide scenarios with high performance and can dynamically adapt and manipulate them as required. Pure rail traffic simulations of bigger areas (e.g. Germany) are also realisable [13].

The free, open source simulation software has the advantage that new algorithms and concepts can be easily tested. Researchers and developers no longer have to implement their own traffic simulation suite, if the potential of commercial simulation tools has been exhausted. In the rail sector, this may be the case when new operating concepts, driving strategies and driving assistance systems are being developed and tested. SUMO can be expanded with new modules at relatively low expense which makes it comparable to other models. Furthermore, freely available source code facilitates cooperation between different institutions, while developing a consolidated simulation platform. SUMO is available under Eclipse Public License V2 and also under the GPL License to provide maximum flexibility for industry and research institutions.

Rail networks can be created and manually edited for use in a SUMO simulation with the help of the graphical network editor which is part of the software suite. Alternatively, networks can also be imported from the OSM, OpenDRIVE, Vissim or Shape-

ter der Gnu Public License V2 lizenziert, um große Flexibilität für industrielle und universitäre Forschung zu bieten.

Schiennetzwerke für die Durchführung von Eisenbahnsimulationen in SUMO können mit einer graphischen Nutzeroberfläche erstellt und bearbeitet werden. Außerdem können Schiennetze aus verbreiteten Datenformaten wie OpenStreetMap (OSM), OpenDRIVE, Vissim und Shapefile mit den Softwaremodulen von SUMO importiert werden. Eine Werkzeugkette zur Verarbeitung des "Spurplan"-Formats der Deutsche Bahn AG existiert ebenfalls, ist aber momentan nicht frei verfügbar.

Bei der Bahnsimulation unterstützt SUMO die klassische blockbasierte Sicherungsinfrastruktur sowie modernere Sicherungskonzepte, die ohne Signalblöcke auskommen. Ein Ausfall der Sicherungssysteme sowie Eisenbahnunfälle können ebenfalls simuliert werden. Für eine multimodale Simulation können außerdem beschränkte und unbeschränkte BÜ modelliert werden. In SUMO werden komplette Züge als Einzelobjekte (d.h. ohne Betrachtung der Wagen) modelliert. Die Flügelung kann dennoch simuliert werden. Die Züge können in Bezug auf ihre Länge, Fahrdynamik und Kapazität (Personen und Fracht) konfiguriert werden. Darüber hinaus kann die Befahrbarkeit für verschiedene Streckenabschnitte durch einzelne Züge beschränkt werden, um beispielsweise nicht-elektrifizierte Strecken zu modellieren.

Ein simulierter Zug tritt zu einem festgelegten Zeitpunkt in das Simulationsgebiet ein und durchfährt dann eine Folge von Gleisabschnitten. Diese Folge kann vorab festgelegt werden oder dynamisch aus einer vorgegebenen Folge von Haltepunkten errechnet werden. Darüber hinaus können Fahrwege durch Simulationsergebnisse oder externen Input modifiziert werden. SUMO enthält bereits Werkzeuge, um einen Zugfahrplan aus OSM-Daten zu erstellen. Die Veröffentlichung von Werkzeugen für den Import von GTFS ist für das Jahr 2020 geplant. Die Unterstützung für das railML-Format ist ebenfalls geplant. Im Gegensatz zum Straßenverkehr wird der Schienenverkehr kontinuierlich von Fahrdienstleitern überwacht, um auf Störungen und Konflikte reagieren zu können. Dieser komplexe Prozess ist bereits in vereinfachter Form in SUMO integriert. Störungsmanagement beliebiger Art kann jedoch als externer Prozess über die TraCI-Schnittstelle integriert werden. Die TraCI-Programmierschnittstelle ermöglicht die Überwachung sowie den Eingriff in eine laufende Simulation über einen Netzwerksocket. Im Bahnbereich kann auf diese Weise das Verhalten von einzelnen Zügen modifiziert werden, um Störungen oder eine Störungsbehandlung zu simulieren. Neben der reinen Fahrzeugsimulation ermöglicht SUMO die Simulation von Personen und ihren intermodalen Reiseketten. Personen können sich als Fußgänger durch die Simulation bewegen und nacheinander verschiedene Fahrzeuge nutzen. Werkzeuge für die Berechnung von Reiseketten zwischen gegebenen Orten sind bereits in SUMO integriert. Dadurch erweitert sich die Möglichkeit, intermodale Verkehrssysteme im Hinblick auf ihre Leistungsfähigkeit zu untersuchen. Durch die Simulation von Personen können Fahrpläne und ein Ereignismanagement aus Sicht der Passagiere betrachtet werden. Die Simulation ermöglicht auch den Blick auf zukünftige bzw. experimentelle Schienenverkehrssysteme und kann somit den Dialog zwischen Forschern, Behörden und Bürgern in einer frühen Planungsphase fördern.

4 Umsetzung

4.1 Middleware

Als Middleware wird nachfolgend jene Software bezeichnet, welche den Datenaustausch und die Datenaufbereitung zwischen der Verkehrssimulation SUMO und dem Big-Data-Cluster über-

file data formats. A toolchain for importing the "Spurplan" data format used by the German railway company Deutsche Bahn AG also exists, but unlike the aforementioned formats, this is not currently available for public use.

SUMO also supports fixed block signalling, as well as modern safety concepts that work without blocks. Safety infrastructure failures and train collisions can also be simulated. Furthermore, railway crossings can be modelled in SUMO for the multi-modal simulation of road and rail traffic. SUMO models trains as individual objects, but includes mechanisms for portion working in a simulation. Trains can be distinguished by the types of tracks they are allowed to use as well as by their dynamic properties, such as their length and passenger and freight capacity.

Each train enters the network at a pre-defined time and then moves through the simulation on a sequence of tracks. This sequence can be pre-defined at the start of the simulation or it can be computed dynamically based on a number of intermediate locations (i.e. stations). Altering the route in reaction to simulation events or external input is also possible. Tools for creating train definitions for public transport based on OSM data are already part of SUMO. Tools for importing GTFS data are scheduled for release in 2020. Support for the RailML format is planned.

Unlike road traffic, rail traffic is continuously supervised by dispatchers, who intervene in case of disruptions and potential conflicts. Modelling this complex process requires a new operational layer that is currently only partially included in SUMO. Complex disposition rules can be simulated by attaching an additional process to a running simulation via TraCI that acts in a supervisory role. The TraCI API enables online interaction with the simulation via a socket connection and the changing of nearly all its values. In the case of rail scenarios, this allows, for example, the modelling of special behaviour for single vehicles or the integration of disruptions.

In addition to rail and road vehicles, SUMO also includes the simulation of people and their intermodal trip chains. People can move through the network as pedestrians or using a vehicle and they can arbitrarily change between the different modes of transport. Several SUMO tools support the intermodal routing and rerouting of vehicles and passengers. This creates value added in relation to the examination of intermodal transportation systems when compared to other simulation tools. In particular, it enables a passenger-centric view of relevant questions such as rail schedule definition and rail disposition in the case of unforeseen events. The simulation results allow the impact of new rail traffic systems to be analysed outside existing system boundaries. This unlocks the potential integration of more stakeholders into the planning, evaluation and optimisation of traffic.

4 Implementation

4.1 Middleware

The middleware tool and SUMO simulation data have been used to reproduce the use cases defined in the project, which has helped in understanding the extended applicability of the use cases, as well as changing the parameters, such as the hopping station position, and evaluating the results. Furthermore, they have also been used to test the ability of the big data cluster to process the large amounts of data pertaining to the use cases, as well as during the initial phase of the project, when the physical setup was under installation. The simulation envi-

nimmt. Die Middleware ist in Python implementiert. Die Middleware übernimmt dabei auch die Ansteuerung der Simulation für die verschiedenen Anwendungsfälle sowie die Verwaltung von Parametervariationen, bspw. die Variation der Position von Hopping-Stationen. Neben der Analyse der Anwendungsfälle dient die Middleware auch zum Testen der Leistungsfähigkeit des Big-Data-Clusters. Letzteres war insbesondere in der Anfangsphase des Projektes vor der Installation der physischen Komponenten im Feld notwendig.

Zusammen mit der Simulation ermöglicht die Middleware eine Nachbildung der bahnseitigen Komponenten, die im Realbetrieb für die Nachrichtenerzeugung sorgen. Dabei leistet die Middleware sowohl Überwachung, Verarbeitung, Serialisierung und Nachrichtentransfer an das Datenmanagementsystem (Bild 2).

Das in SUMO simulierte Verkehrsnetz besteht aus dem deutschen Schienennetz (Bild 2) sowie aus kleinen Ausschnitten des Straßennetzes an jedem vorhandenen BÜ (Bild 3). Die kleinen Ausschnitte aus dem Straßennetz werden verwendet, um den Anwendungsfall „Anrufschränke“ abzubilden. Die Middleware verwaltet alle Simulationsparameter wie Simulationsdauer, Eingabedateien für Zugrouten, Haltepunkte usw. Die bahnseitige Infrastruktur wie z. B. Gleisgeometrie, Gleisstopologie, BÜ und Weichen, welche für die Anwendungsfälle benötigt werden, entstammen aus der OSM-Datenbank und sind Teil des Simulationsnetzes. Dabei gibt es entsprechend jeder Weiche und jedes BÜ einen dazugehörigen Netzknoten. Dies wurde in der Simulation so umgesetzt, um alle Infrastrukturelemente aus dem Anwendungsfall „Service und Diagnose“ in einem bestimmten Radius zu gruppieren und über eine V2X-Kommunikationseinheit abzubilden. Wetterstationen, die in der Simulation vorhanden sind, werden als künstlicher Punkt für jede räumlich benachbarte Gruppe von Weichen als mittlerer Punkt berechnet. Das Bahnnetz wurde über die Overpass-Schnittstelle [14] aus der OSM-Datenbank erstellt. Bei der Erzeugung des Straßennetzes an den BÜ war es notwendig, dieses für jedes Bundesland einzeln zu importieren. Grund dafür war, dass die Anfrage an die Schnittstelle für das gesamte Gebiet zu rechnerintensiv gewesen wäre. Die einzelnen Netzteile wurden anschließend mit dem Bahnnetz zu einem Gesamtnetz vereinigt. Die Middleware baut intern eine Repräsentation aller Systemkomponenten (Züge, Weichen, BÜ, Haltepunkte, ...), welche dann für die Erzeugung der Nachrichten untereinander sowie mit den Simulationsdaten verknüpft werden (Bild 3).

In der Middleware wird jeder Zug während jedes Schrittes des Simulationsablaufs verfolgt. Dabei wird fortlaufend geprüft, ob ein Zug sich im Kommunikationsradius von V2X an den Hopping-Stationen, Bahnhöfen, BÜ sowie am Mittelpunkt von gruppierten Weichen befindet. Sobald ein Zug in den Radius eintritt bzw. diesen verlässt, wird eine Ereignisnachricht erzeugt und abgespeichert. Wenn ein Zug an einem Bahnhof anhält, wird ebenfalls eine Nachricht erzeugt. Die Speicherung der Nachrichten erfolgt dabei direkt oder auch gepuffert in einer HDF5-Datei. Das „Hierarchical Data Format“ (HDF) [15] ist ein offenes Datenformat für die Speicherung großer, heterogener Datensätze in einer verzeichnisartigen Struktur. Eine sogenannte HDF5-Gruppe wird dabei für jeden Zeitschritt der Simulation erzeugt. Diese Gruppe wiederum beinhaltet alle Nachrichten dieses Zeitschritts. Die Daten aus der HDF5-Datei werden nachfolgend in den Kafka-Cluster geschrieben.

Die Verarbeitung der Daten in der Middleware unterscheidet zwischen Bewegungsdaten „OGC Moving Feature“ (MF) und ortsbezogenen Messdaten „OGC Observation and Measurements“ (OM) [10]. Die MF-Nachrichten enthalten die Trajektorien von Zügen und Straßenfahrzeugen, die sich im Rail2X-Radius um einen

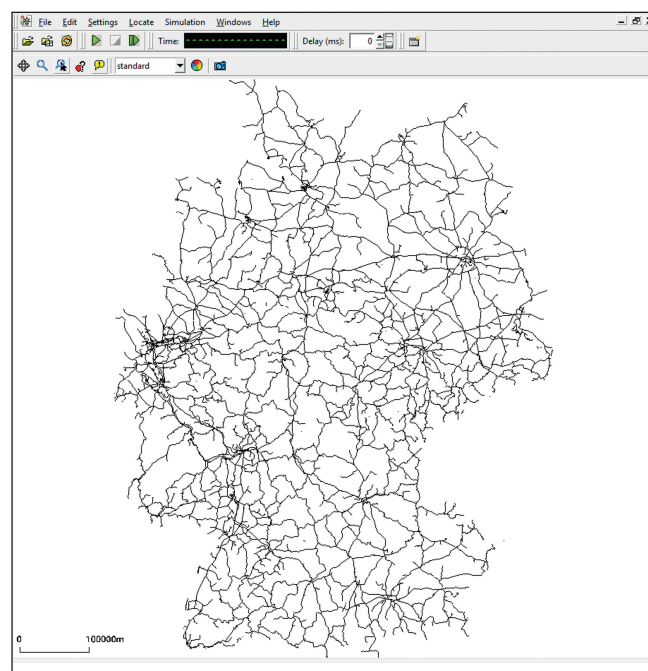


Bild 2: Das deutsche Bahnnetz aus OSM in der graphischen Darstellung von SUMO

Fig. 2: The German railway network as seen in SUMO

Quelle / Source: SUMO

ronment imitates the scenarios responsible for generating messages in a real-world railway network. The middleware built in this project undertakes the monitoring, processing, serialising and pushing of the messages generated in the simulation to the data management server. It has been developed using Python programming and acts as a connecting bridge between SUMO and the cluster.

The SUMO simulation network consists of railway tracks across Germany (fig. 2) and large number of small road clusters connected to every level-crossing in the railway network (fig. 3). The small road clusters are used to reproduce the second use case “Level Crossing”. The middleware manages all the parameters involved in the simulation, including the number of time steps, the look-up files for the route, the stops, the railway stations, etc. Other additional railway infrastructure elements, such as track geometry, track topology, switches and level-crossings that are required to reproduce the use cases are added to the simulation network explicitly from OSM database. These nodes replicate the actual switch and level-crossing locations where signals are generated. This is done in order to group all the infrastructure elements from the “Service and Diagnosis” use case within a certain radius and to connect them via a V2X communication unit. The representation of weather stations is realised in the simulation as a pseudo point that is generated by taking the mean coordinate of all the switches in a cluster. A switch cluster is a group of switches that is located within a specific threshold distance. The railway network has been produced using the overpass queries [14] on the OSM database. In order to overcome the limits of the computational resources to generate the network, the road clusters have been generated separately for each federal state of Germany and then combined with the railway network. Programmatically, every component (trains, cars, switches, level-crossings, railway stations, etc.) in the simulation is represented as an individual class, all of which are then interconnected to form the complete middleware.

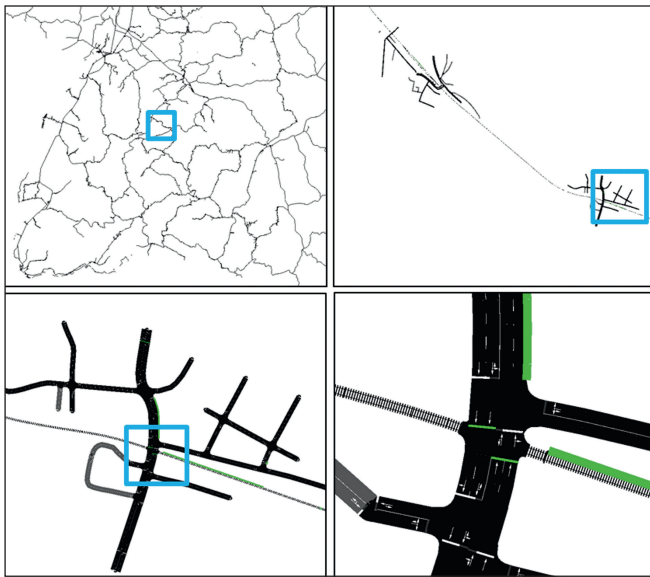


Bild 3: Visualisierung eines BÜ in SUMO

Fig. 3: A view of a LX in SUMO

Quelle / Source: SUMO

BÜ, eine Weiche oder einen Bahnhof bewegen. Wetter- und sonstige Telemetriedaten werden als OM-Nachrichten übermittelt. Da jedoch OM-Daten nicht von SUMO erzeugt werden, erfolgt die Erstellung der entsprechenden Nachrichten über die Middleware. Neben den eigentlichen Nutzdaten enthalten alle Nachrichten noch Metadaten wie Fahrzeug-ID, Orts-ID und Nachrichtentyp. Die Simulationsnachrichten, die in HDF5 gespeichert wurden, werden in der zeitlichen Reihenfolge der Simulation abgespielt und an das Big-Data-Cluster übertragen. Diese Nachrichten werden über ein Python-Skript (Kafka Producer) durch POST-Anfragen versendet.

4.2 Datenverarbeitung im Big-Data-Cluster und Datenvisualisierung

Bild 4 stellt den Workflow der Datenverarbeitung im Big-Data-Cluster dar. Das Big-Data-Cluster erhält die Nachrichten von Infrastrukturelementen und Fahrzeugen sowohl von der Middleware als auch von aufgebauten Systemen im Feld über die gleiche Schnittstelle. Diese Nachrichten sind im Kafka-Cluster unter zwei Topics gespeichert. Das erste Topic enthält alle Nachrichten im OGC-MF-Format und das andere enthält alle Nachrichten in OGC-OM-Format. Zwei Flink-Anwendungen laufen rund um die Uhr, um die Nachrichten aus dem Datenstrom zu verarbeiten. Die erste Flink-Anwendung liest die Trajektorien der Straßenfahrzeuge und Züge aus und speichert sie in einer HBase-Tabelle mit Zeitstempel, ID des Fahrzeuges und Quelle der Nachricht (Feld oder Simulation). Die zweite Flink-Anwendung trägt in einer HBase-Tabelle eine Zeile für jede angekommene Nachricht ein. Diese Zeile enthält die ID des Objekts, das die Nachricht generiert hat, die ID der Station, von der die Nachricht übermittelt wurde, die Größe der Nachricht und zeitliche Eigenschaften. Die Überquerungsanfragen von Straßenfahrzeugen, der Zustand der Bahnschranke und Bedarfshaltanfragen der Züge sind ebenfalls in dieser Tabelle für weitere Analysen gespeichert. Die OM-Nachrichten mit den „Service und Diagnosedaten“, wie den Daten aus der Wetterstation, werden auch in einer Tabelle in HBase abgelegt. Alle Flink-Anwendungen sind in Java geschrieben und benutzen die Java-Bibliotheken von Kafka, Flink, HBase, GeoTools [16] und GeoMesa. GeoMesa-HBase-Datenspeicher werden auf dem GeoServer be-

The middleware establishes a virtual environment for generating the event messages. This is achieved by tracking every moving element at every time instance of the simulation. The middleware then calculates distances for each moving vehicle after regular time intervals and checks, if any of them satisfy the location proximity conditions such as crossing inside or outside the range of V2X at the hopping stations, railway stations, LC and the mid-point of the switch clusters. If the condition is met, the corresponding event signal is generated and saved either to an HDF5 file or in the program variables that are later saved to the HDF5 file. Some events are also generated when trains have stopped at a railway station. The Hierarchical Data Format (HDF) [15] is an open-source data format that can be used to store large heterogeneous datasets organised in a directory-like fashion. An HDF5 group is created for each time step in the simulation and the messages generated during the simulated time period are stored in corresponding groups as HDF5 Datasets. The messages generated during the simulation are first saved to an HDF5 file and later pushed into the Kafka cluster.

The processing of these messages is different based on their broad types, namely the OGC Moving Feature (MF) and OGC Observation and Measurements (OM) [10]. The MF messages contain the trajectories of the trains and cars as they travel across the Rail2X range at a railway station, level-crossing and switch cluster. Weather data, telemetry and logging data from the infrastructure elements is stored in OM messages. However, these messages are filled with synthetic data, because SUMO does not actively generate OM data. All the signals are stored with details such as the vehicle-ID, location-ID and type of signal along with the actual payload content. The signal messages stored in the HDF5 file throughout the entire course of the simulation are serialised according to the time step when they were processed. Arranging the messages by time step helps to forward them to the server in the same sequence and to adhere to the movement of the vehicles in the network. Finally, the processed messages are picked up by a python script (Kafka Producer) which pushes all the messages in the HDF5 file to the big data cluster using POST calls.

4.2 Data processing in the big data cluster and data visualisation

Fig. 4 shows the data processing workflow in the big data cluster. The messages sent from the middleware described in the previous section are received by the big data cluster via an interface. This interface is common for both the simulation and the units present in the field. These messages are stored in the Kafka cluster under two different topics. One topic contains all the messages in the OGC Moving Features format, while the second topic contains all the messages in the OGC Observations and Measurements format. Two Flink applications (Flink-Kafka consumers) run around the clock in order to process the messages from the data streams. The first Flink application extracts the trajectories of the cars and trains and stores them in an HBase table along with the timestamps, the vehicle identity and the message source (field or simulation). The second Flink application stores one record for each message received from the uploading stations in an HBase table. These records consist of the ID of the object which generated the message and the station from which the message was received, the size of the message and the temporal properties. As such, the requests made by cars to open a LC and the requests to stop a train are stored in this table for further analysis. In the case of OM messages containing weather data

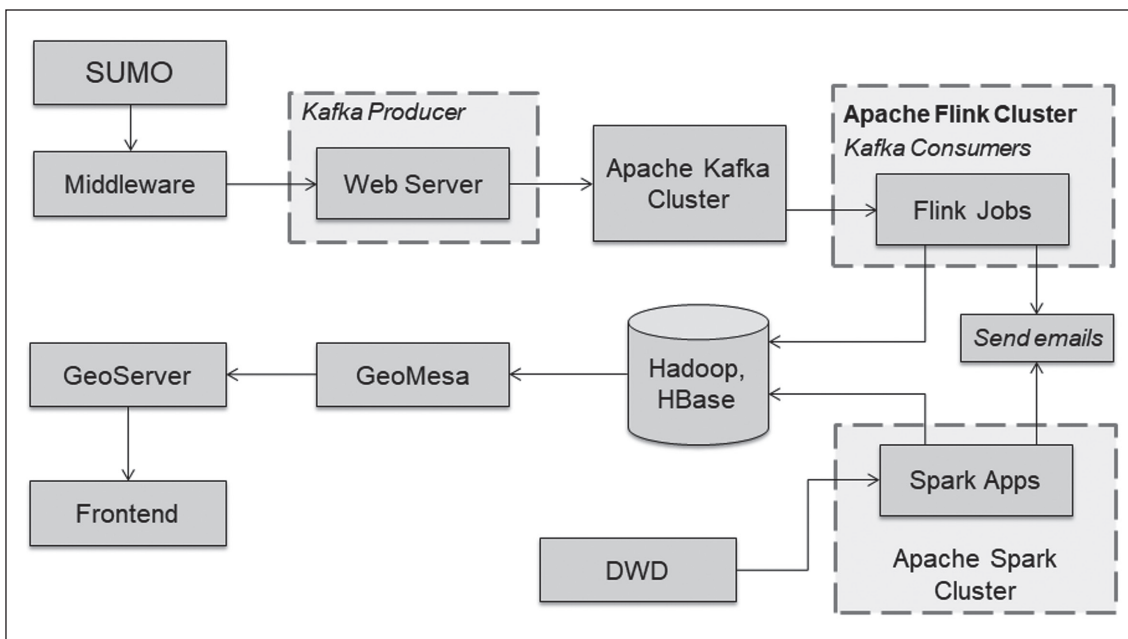


Bild 4: Workflow im Big-Data-Cluster

Fig. 4: The workflow in the big data cluster
Quelle / Source: DLR

reitgestellt, um abgespeicherte Daten aus HBase über OGC-standardisierte Schnittstellen bereitzustellen (Bild 4).

Eine dritte Flink-Anwendung validiert die empfangenen Wetterdaten von der im Projekt aufgebauten Wetterstation sowie von den in der Simulation synthetisch generierten Wetterstationen und gewährleistet die Validierung der Wetterdaten in Echtzeit. Es wird überprüft, ob der Wert des Temperaturkanals realistisch ist. Wenn der Wert außerhalb eines bestimmten Temperaturbereichs liegt, wird eine Zeile in einer HBase-Tabelle mit entsprechenden Metadaten eingefügt. Eine E-Mail mit Warnungen über mögliche falsche Messwerte der Wetterstationen wird an eine vordefinierte Liste von Empfängern versendet.

Eine Spark-Anwendung wurde ebenfalls entwickelt, um die Konsistenz der übermittelten Wetterdaten durch den Vergleich mit Wetterdaten vom Deutschen Wetterdienst (DWD) zu prüfen. Dafür speichert eine zweite Spark-Anwendung jeden Tag die Wetterdaten vom DWD-FTP-Server [17] ins HDFS ab. Die Koordinaten der DWD-Wetterstationen, der SUMO-Wetterstationen und die der Wetterstation im Erzgebirge sind in HBase-Tabellen gespeichert. Die DWD-Stationen in der Umgebung jeder Wetterstation in der Simulation und im Feld werden durch einen GeoMesa-WPS-Vorgang identifiziert und mit Wetterdaten des DWD für den gleichen Zeitraum verglichen. Die GeoTrellis-Scala-Bibliothek [18] wird zur Geodatenverarbeitung in der Spark-Anwendung verwendet. Diese Spark-Anwendung läuft in einem bestimmten Zeitintervall und schickt einen Bericht mit allen identifizierten Fehlern an eine vordefinierte Liste von Empfängern. Neben der Validierung in Echtzeit über die Flink-Anwendung führt die Spark-Anwendung eine Konsistenzprüfung der Wetterdaten durch. Beide Spark-Anwendungen wurden in Scala geschrieben. Über die graphische Oberfläche von Grafana wird die Leistung des Kafka-Producers und Kafka-Consumers überwacht. Die Bild 5(a) zeigt die Rate der eingehenden Nachrichten in Kafka, Bild 5(b) wiederum zeigt die Rate der verarbeiteten Nachrichten in den Flink-Anwendungen. Diese Diagramme wurden von einer Simulation erzeugt, die 1000 Sekunden (16,67 Minuten) lief und dabei 915 000 Nachrichten erzeugt hat. Die Metriken, die jede Komponente des Big-Data-Clusters senden, sind umfangreich. Eine Liste der Metriken befindet sich in der Prometheus-Dokumentation [11]. Durch die Visualisierung der Metriken in Grafana war es

from the “Service and Diagnosis” use case, the observed weather parameters are also stored in an HBase table. These Flink applications are written in Java and use the Java libraries from Kafka, Flink, HBase, GeoTools [16] and GeoMesa. GeoMesa-HBase data stores are created on the GeoServer and connect to these HBase tables. Various layers are published on the GeoServer and served using OGC standardised services.

A third Flink application validates the weather data received from the weather station established in the field, as well as the weather stations in the simulation that generate synthetic data. This ensures the validation of the recorded weather data in real time. It tests whether the recorded temperature values are realistic using constant boundary conditions. When a value departs from the defined bounds, an entry is made in an HBase table along with the relevant metadata. Furthermore, an email with warnings about weather stations generating these erroneous data entries is sent to a predefined set of recipients.

A Spark application is also being developed to verify the consistency of the recorded weather data by comparing it with the values from the Deutsche Wetterdienst (DWD). For this purpose, a second Spark application runs once a day and downloads weather data from the FTP Server from DWD [17] and saves it into Hadoop HDFS. The locations of the DWD stations as well as the weather stations from the SUMO simulation and the Erzgebirgsbahn have been stored in an HBase table. The DWD stations around each weather station in the simulation and the field are identified using processes available in GeoMesa and the recorded weather values are compared with those from the DWD for the same time period. The GeoTrellis Scala library [18] is used to handle the geospatial data within the Spark application. This application runs at specific time intervals and sends reports of any inconsistent values to a predefined set of recipients. This Spark application thus performs a consistency check on the weather values in addition to the real-time validation performed by the Flink application. Both Spark applications are written in the Scala programming language. The performance of the Kafka producer and consumers as well as the Spark applications is monitored using the Grafana web

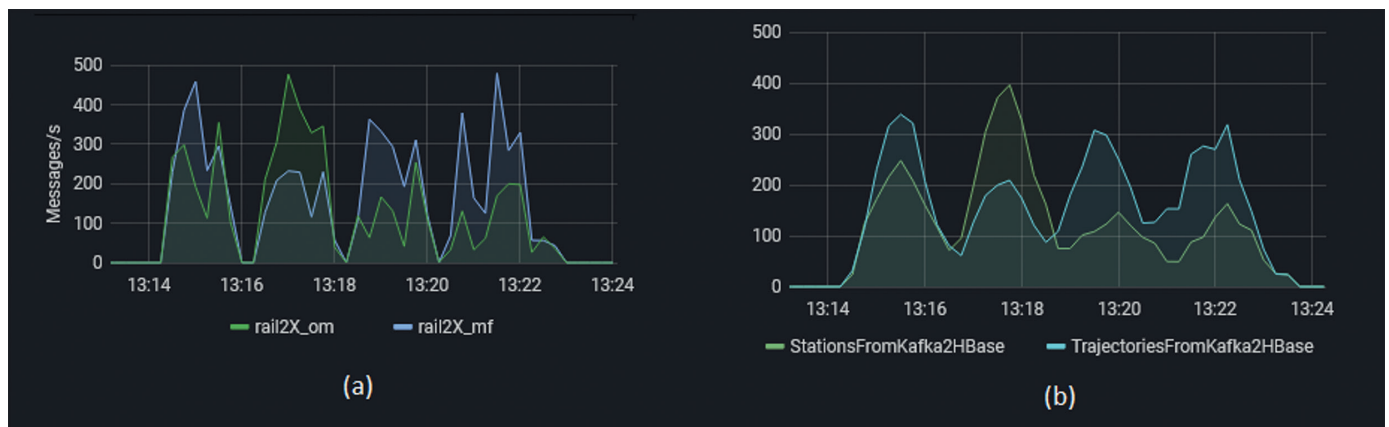


Bild 5: (a) Rate der eingehenden Nachrichten aus der Simulation ans Kafka-Cluster, (b) Rate der verarbeiteten Nachrichten in Flink-Anwendungen

Fig. 5: (a) The rate of the messages sent from the middleware to two topics in the Kafka cluster, (b) the rate of the messages from the two topics processed by the Flink jobs

Quelle / Source: DLR

möglich, die Anzahl an Partitionen pro Topic in Kafka zu optimieren, sodass die Parallelisierung der Flink-Anwendungen optimiert werden konnte. Dadurch können die Nachrichten schnell und mit geringster Latenz verarbeitet werden, was zur optimalen Nutzung der Ressourcen führt. Das verbessert die Echtzeit-Visualisierung von Fehlermeldungen im Frontend.

Die ersten prototypischen Analysen der Simulationsdaten wurden durchgeführt und können in der Frontend-Anwendung des Datenmanagementsystems „Bahnsystem“ [19] visualisiert werden. Eine Analyse davon ist die aggregierte Größe der übermittelten Nachrichten von Stationen in der Simulation. Das Ergebnis dieser Analyse wird auf der Karte visualisiert (Bild 6(a)). Diese Visualisierung ist nützlich, um ein Verständnis über die Verteilung der Nachrichten in unterschiedlichen Stationen in verschiedenen Zeitfenstern in Deutschland zu erlangen. Die Visualisierung von fehlerbehafteten Wetterdaten pro Station, die im Validierungsprozess identifiziert wurden, wird aggregiert auf einer Karte visualisiert (Bild 6(b)). Diese Visualisierung ergänzt die textuellen Berichte, die während des Validierungsprozesses über Flink- und Spark-Anwendungen generiert wurden. Die Fahrzeugdichte in verschiedenen Abschnitten des Verkehrsnetzes in unterschiedlichen Zeitfenstern kann durch die Generierung einer Heatmap, basierend auf den Trajektorien der Straßenfahrzeuge und Züge, dargestellt werden. Ein Beispiel dafür bietet die Visualisierung in Bild 6(c). Durch GeoMesa ist es möglich, die Heatmaps von tausenden von Trajektorien in wenigen Sekunden zu berechnen und zu visualisieren, sodass diese in Echtzeit visualisiert werden können. Eine Erreichbarkeitskarte kann für jeden Bahnhof und BÜ in der Simulation berechnet werden. Ein Beispiel wird in Bild 6(d) dargestellt. Die „Service und Diagnose“-Daten, die von Infrastrukturelementen aus diesem Anwendungsfall erfasst werden, werden durch einen Zug transportiert und an den Bahnhöfen und BÜ an das Big-Data-Cluster übermittelt. So ergibt sich eine Differenz zwischen dem Zeitpunkt, an dem der „Service und Diagnose“-Datensatz – wie beispielsweise Wetterdaten – an den Zug übersendet wurde, und dem Zeitpunkt, an dem diese Nachricht ans Big-Data-Cluster übermittelt wurde. Diese Erreichbarkeitskarten wurden durch die Interpolation der Latenzwerte generiert. Auf Basis dieser Informationen wurde eine Barnes-Surface-Interpolation [20] durchgeführt. Mit diesen Karten können jene Abschnitte des Bahnnetzes identifiziert werden, die weit von den Stationen des Big-Data-Clusters entfernt sind. Dies ist hilfreich, um die Positionierung von Stationen und dazugehörigen Hopping-Stationen zu optimieren.

interface. Fig. 5(a) shows the rate of the messages coming into the data streams in Kafka and 5(b) shows the rate of the message processing by the Flink applications. These plots have been generated from a simulation dataset consisting of about 915,000 messages from 1000 time steps which corresponds to 16.67 minutes. The metrics reported to Prometheus by each component in the big data cluster are extensive and the list can be found in the Prometheus documentation [11]. It was possible to optimise the number of partitions for each Kafka topic using the plots of these parameters in Grafana so that the parallelism for the Flink applications could also be optimised to achieve fast message processing with the least latency and optimal resource consumption. This has facilitated the frontend visualisation and error notification in real-time.

The initial analysis of the data from the simulation has been realised and the results have been visualised in a web-frontend application available in the „Bahnsystem“ data management system [19]. One of them is the cumulative size of the messages received from the different stations, generated from the simulation dataset and visualised on the map (fig. 6(a)). This helps in understanding the distribution of the messages across the stations in Germany during different time periods. The visualisation of the stations generating erroneous weather data as identified during the weather data validation process (fig. 6(b)) has also been generated. This visualisation complements the textual error reports generated during the weather data validation by the Flink and Spark applications. The density of the vehicles driving along different segments of the network in different time periods has been visualised through the generation of a heat map of the trajectories from the cars and trains (fig. 6(c)). The capability of the GeoMesa processes to generate heat maps from thousands of trajectories in few seconds means they can be visualised in real time. Reachability maps have been generated for each railway station and LC; one such map is shown in fig. 6(d). Since the data generated from the various infrastructure elements on the network for the „Service and Diagnosis“ use case has only been deposited in the big data cluster at the railway stations and LC, there is a difference in the time between the message generation at these entities on the network and the reception of the messages in the big data cluster. The reachability map for each station is the result of the interpolation of these latency values. Barnes surface interpolation [20] has been performed on the input

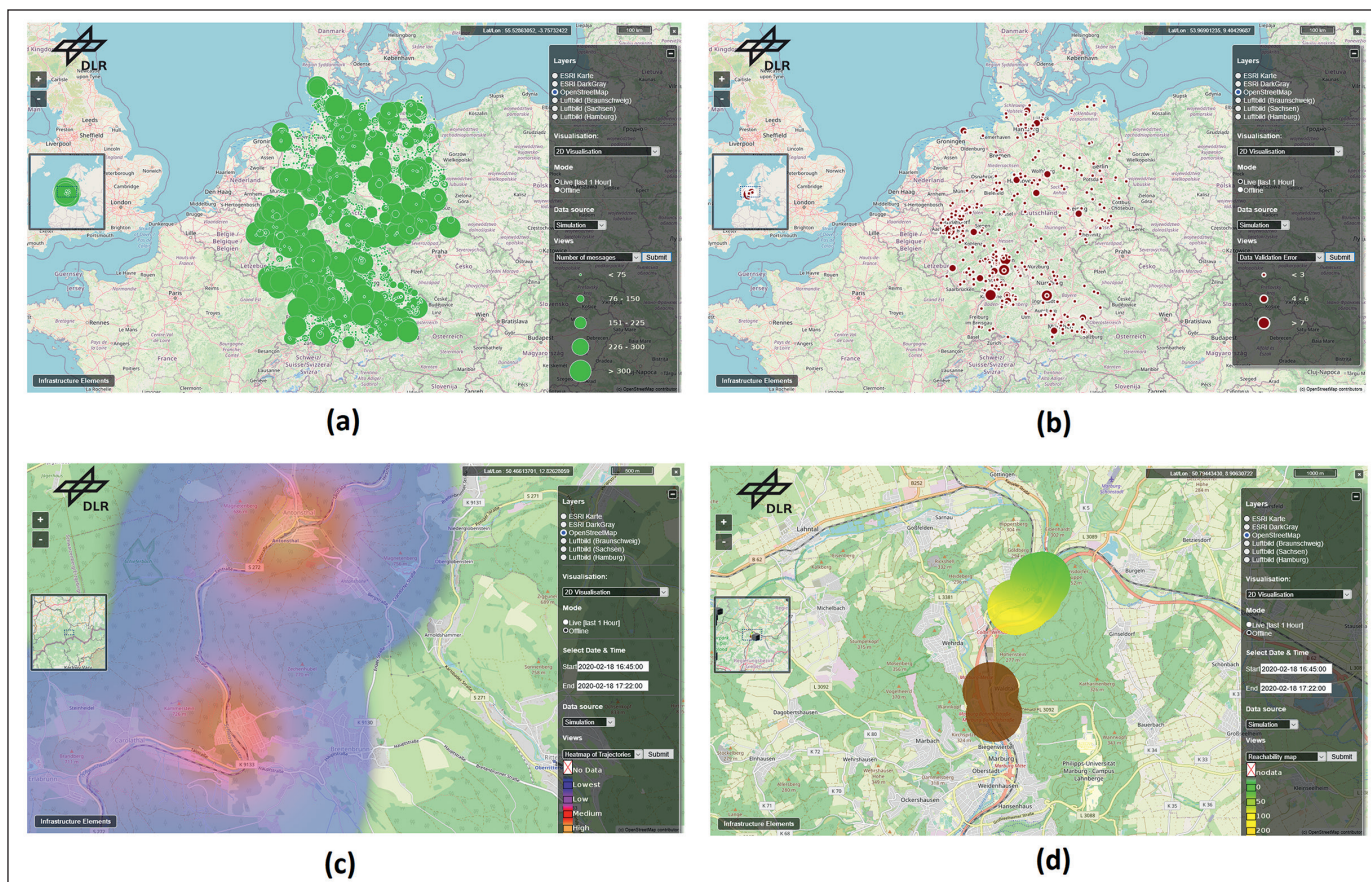


Bild 6: (a) Aggregierte Größe der übermittelten Nachrichten von Stationen in Deutschland, (b) Wetterstationen, die falsche Wetterdaten generiert haben, (c) Heatmap der Trajektorien von Fahrzeugen in dem Gebiet um Antonsthal, (d) Erreichbarkeitskarte für Cölbe Bahnhof.

Fig. 6: (a) The cumulative size of the messages received from the stations across Germany (b) The stations generating erroneous weather data identified during validation (c) A heat map of the trajectories from the vehicles recorded in the region around Antonsthal (d) The reachability map for the Cölbe railway station.

Quelle / Source: DLR

5 Fazit und Ausblick

Dieser Beitrag beschreibt die Nutzung von Open-Source-Big-Data-Technologien in der Prozessierung von aufgenommenen Daten von Infrastrukturelementen und Fahrzeugen in Zusammenhang mit den drei Anwendungsfällen aus Rail2X. Die SUMO-Simulationsanwendung wird verwendet, um die Bewegung von Zügen und Straßenfahrzeugen im Schienen- und Straßennetz in ganz Deutschland zu simulieren. Durch die Middleware-Anwendung zwischen SUMO und dem Big-Data-Cluster werden die Straßenfahrzeug- und Zugbewegungen aus SUMO erfasst und prozessiert, die Anwendungsszenarien abgeleitet und die Nachrichten an das Big-Data-Cluster übermittelt. Durch die großskalierte Simulation der Anwendungsfälle war es möglich, erste prototypische Ergebnisse der Extrapolation der Anwendungsfälle zu analysieren und zu visualisieren sowie einen ersten Überblick über die Möglichkeiten der Echtzeitprozessierung großer Datenmengen im Big-Data-Cluster zu erhalten. Die ersten Tests des Big-Data-Clusters wurden mit Simulationsdaten durchgeführt, was nach heutigem Stand eine Prozessierung von ca. 9000 Nachrichten pro Sekunde ermöglicht.

Im nächsten Schritt wird an der Weiterentwicklung von Analysefunktionalitäten, Verbesserung der Echtzeitfähigkeit und Performance-Erhöhung gearbeitet. Für die Abbildung und Analyse neuer Anwendungsfälle spielen Datensätze wie Messdaten, Kartenmaterial und Fahrpläne eine sehr wichtige Rolle. Aus diesem

values to generate the reachability maps. These maps are helpful in identifying sections of the network which are far from data uploading stations, which in turn could be used to optimise the positions of these stations, as well as the placement of the hopping stations.

5 Conclusion and outlook

This work demonstrates the use of open-source, big data technologies to manage data collected from infrastructure elements and vehicles on a railway network related to three Rail2X use cases. The SUMO simulation software has been used to simulate the movement of trains and cars in the railway and road network throughout Germany. The middleware application between SUMO and the big data cluster has been used to read and process the movement of the cars and trains so that the use case scenarios are reproduced and the generated messages are sent to the big data cluster. It has been possible to analyse and visualise the system's initial outputs by running the use cases over a Germany-wide simulation. A first glance of the big data cluster's real-time processing capabilities was also acquired. The initial tests of the big data cluster were performed using simulated data, which enables the processing of 9000 messages/second under current conditions.

The next step involves the further development of the analytical capabilities, as well as increasing the real-time and perfor-

Grund wird für die Weiterentwicklung des gesamten Systems viel Wert auf die Verwendung von standardisierten Datenmodellen, Formaten und Schnittstellen gesetzt, was zur einfacheren Integration mit anderen Systemen beitragen soll. Ein möglicher, zukünftiger Anwendungsfall ist die Integration mit anderen „Echtzeitsystemen“, um komplexe Szenarien aus Simulationsdaten und Realdaten, sogenannten hybriden Szenarien, großskaliert zu prozessieren und zu analysieren.

[Rechtlicher Hinweis / Förderhinweis]

Die hier vorgestellten Arbeiten wurden durch das Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) im Programm mFund gefördert. Die vorliegenden Ergebnisse sind im Rahmen des Projektes Rail2X-„Smart Services“ (19F2010) entstanden. ■

mance capabilities. Measurement data, maps and driving schedules play an essential role in the establishment and analysis of new use cases. The use of standardised data models, formats and interfaces to facilitate integration with other systems will therefore be very valuable. A possible future use case for the system involves the integration of our system with other “real time systems” in order to process and analyse complex scenarios from simulated and real data, which are also called hybrid scenarios.

[Legal notice]

The work presented here was funded by the Federal Ministry of Transport and Digital Infrastructure (BMVI) under the mFund program. The results presented here are the result of the Rail2X-Smart Services project (19F2010). ■

LITERATUR | LITERATURE

- [1] <https://elib.dlr.de/127084/>, letzter Zugriff 23. Juni 2020 um 13:00
- [2] <https://www.openstreetmap.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [3] <https://kafka.apache.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [4] <https://flink.apache.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [5] <https://spark.apache.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [6] <https://hadoop.apache.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [7] <https://hbase.apache.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [8] <https://www.geomesa.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [9] <http://geoserver.org/>, letzter Zugriff 13. Mai 2020 um 09:00
- [10] <https://www.ogc.org/standards/>, letzter Zugriff 13. Mai 2020 um 09:00
- [11] <https://prometheus.io/>, letzter Zugriff 13. Mai 2020 um 09:00
- [12] <https://grafana.com/>, letzter Zugriff 13. Mai 2020 um 09:00
- [13] Wießner, E.; Erdmann, J.; Flamm, L.; Jäger, B.: Auf Straße und Schiene mobil – intermodale Verkehrssimulation mit SUMO, SIGNAL+DRAHT, (110) 11/2018, S. 49-55, DVV Media Group. ISBN ISSN 0037-4997
- [14] <http://overpass-api.de/> Overpass-API-Dokumentation, letzter Zugriff 16. Juni 2020 um 12:00
- [15] <https://www.hdfgroup.org/solutions/hdf5/>, letzter Zugriff 16. Juni 2020 um 13:00
- [16] <https://geotools.org/>, letzter Zugriff 10. Juli 2020 um 13:00
- [17] ftp://opendata.dwd.de/climate_environment/CDC/observations_germany/climate/, letzter Zugriff 02. Juni 2020 um 15:00
- [18] <https://geotrellis.io/>, letzter Zugriff 10. Juni 2020 um 12:00
- [19] <https://elib.dlr.de/121116/>, letzter Zugriff 23. Juni 2020 um 13:00
- [20] <https://docs.boundlessgeo.com/suite/1.2.0/cartography/rt/barnes.html>, letzter Zugriff 16. Juni 2020 um 13:00

AUTOREN | AUTHORS

Sangeetha Shankar, M.Sc.

Wissenschaftliche Mitarbeiterin / *Research Associate*
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Verkehrssystemtechnik
Anschrift/Address: Lilienthalplatz 7, D-38108 Braunschweig
E-Mail: sangeetha.shankar@dlr.de

Lucas Andreas Schubert, M.Sc.

Gruppenleiter Datenmanagement und Geodatenverarbeitung /
Team Leader Spatial Data Processing and Engineering
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Verkehrssystemtechnik
Anschrift/Address: Lilienthalplatz 7, D-38108 Braunschweig
E-Mail: lucas.schubert@dlr.de

Akhil Jayant Patil, B.E.

Hilfswissenschaftler / *Student Research Assistant*
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Verkehrssystemtechnik
Anschrift/Address: Lilienthalplatz 7, D-38108 Braunschweig
E-Mail: akhil.patil@dlr.de

Dr. Jakob Erdmann

Wissenschaftlicher Mitarbeiter / *Research Associate*
Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Institut für Verkehrssystemtechnik
Anschrift/Address: Rutherfordstraße 2, D-12489 Berlin
E-Mail: Jakob.Erdmann@dlr.de